DOCUMENT RESUME

ED 146 929 IR 005 400

AUTHOR Sindermann, Robert, Jr.
TITLE Compiling an Index with the Help of a Computer.
PUB DATE 9 Dec 77
NOTE 43p.

EDRS PRICE MF-$0.83 HC-$2.06 Plus Postage.
DESCRIPTORS *Automatic Indexing; *Computer Programs; Flow Charts; *Indexes (Locaters); *Program Descriptions
IDENTIFIERS *BASIC Programing Language; *DEC 10 Computer

ABSTRACT

This report presents a procedure for compiling a typed index using a DECsystem 10 computer without index cards, with the following objectives: (1) to increase compilation speed of a typed index; (2) to provide a machine-readable version of an index that can be edited rapidly; (3) to provide two forms of an index--one arranged by page number to check terms, and the final product arranged by index terms; and (4) to enable an indexer to make several copies. It explains new programs and procedures that can be used by any indexer with access to DECsystem 10 installation. Use of the SOS text editor to create master files, the SORT Program, and two BASIC language computer programs to compile index terms and arrange them in index format are explained. Some fundamental DECsystem 10 monitor level and SOS commands are described and illustrated with examples. Limitations on the content of master files are covered, and steps for the correction of common input errors are detailed. Annotated copies and flow charts of the BASIC programs are included. (Author/KP)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. Nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED146929

U S DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

COMPILING AN INDEX WITH THE HELP OF A COMPUTER

Robert Sindermann, Jr.

Graduate School of Library Science
University of Texas at Austin
9 December 1977

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Robert P. Sindermann

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC) AND USERS OF THE ERIC SYSTEM."

IR005400

Submitted by Mr. E. B. Jackson
to the ERIC Clearinghouse on
Information Resources

PREFACE

The programs and procedures described in this booklet were put together out of necessity. Three of us (Daraka Cook, Kathy Williamson, and myself) in an indexing group at the University of Texas Graduate School of Library Science decided that we would not use index cards at all but instead would try our hand at working with a computer to compile some typed indexes. We believed that we could use the Key-Word-in-Context indexing program available on the University of Texas DECsystem-10 computer to produce a rough draft of an index that we could easily edit down to a conventional format.

After the first few sessions of editing the rough draft, however, we decided otherwise. We concluded that one should use the KWIC indexing program only when one wants a KWIC index and nothing more. The conversion of a detailed KWIC index to a conventional format proved too laborious. The use of index cards would clearly have been more efficient, even if we had had to type the index after the cards were put in final order.

At that point we were in a dilemma: on the one hand, we would have lost several weeks' worth of work had we started over with cards; and on the other, we would have gone on spending painful hours doing the extensive editing required to make the KWIC index look presentable.

Necessity was once again the mother of invention. This

student had become familiar enough with programming in the BASIC language to want to salvage the indexing project by devising a program that would produce an index in a conventional format--an index that would not require any editing just to make it readable.

Some initial attempts were made to run a program that would perform three operations: the compiling, the sorting, and the formatting of index terms, their modifications, and their page references. It was soon found, however, that the sorting capability of the BASIC language is much less efficient than that of a DECsystem-10 program. But since no DECsystem-10 program can perform the other two operations satisfactorily, two specialized programs were written in BASIC to take care of them. A procedure in three parts has thus been worked out and has been tested and found useful in two indexing projects undertaken in the fall of 1977.

While the descriptions of commands and procedures in this booklet are based on experience with the particular versions of software that are implemented on the University of Texas computer system, they have been made general enough to apply to most, if not all, DECsystem-10 installations. Software variations among installations are bound to occur, though, and users are all well advised to consult local system personnel about any important ones that affect them before they begin their indexing projects.

The indexing procedure is described here in the hope that other students who work on indexing projects and who have

access to DECsystem-10 installations will find it useful, too. And if anyone cares to study my BASIC programs for their own sake, perhaps with his mind set on improving them, flowcharts and annotated copies of them are included at the end of this booklet.

Robert Sindermann, Jr.

Graduate School of
Library Science

University of Texas
at Austin

9 December 1977

CONTENTS

Introduction 1

DECsystem-10 monitor-level commands 5

Using SOS to create a master file 8
- Common SOS commands 9
- SOS Alter mode procedures 14
- Exiting from SOS 15

Requirements for the master file 17
- Suggested punctuation conventions 19
- Eliminating errors of spacing in master file lines . 20

Processing the master file 22
- Using the BASIC compiling program 22
- Using the DECsystem-10 SORT program 24
- Using the BASIC formatting program 25
- Using SOS to replace the punctuation conventions . . 28

Ordering a printed copy of the index 29

ARCHIV.BAS, annotated 30
Flowchart of ARCHIV.BAS 32

FORMAT.BAS, annotated 33
Flowchart of FORMAT.BAS 36

COMPILING AN INDEX WITH THE HELP OF A COMPUTER

INTRODUCTION

This booklet presents a procedure for compiling a typed index with the aid of a DECsystem-10 computer such as the one at the University of Texas at Austin. The procedure was designed with these objectives in mind:

1) to help an indexer to compile a typed index more quickly than he could otherwise do by using index cards;

2) to provide an indexer with a machine-readable version of his index that can be edited more rapidly than can a product of the DECsystem-10 Key-Word-in-Context indexing program;

3) to provide an indexer with two forms of his index--one form (the original, master file of index terms drawn from the text) arranged by page number, to allow him to **check** the index terms, page number by page number, against the notes he has made in the printed text being indexed, and the other form (the final product) arranged by index term;

4) to enable an indexer to easily make several copies of **an** entire index for distribution, for instance, to other members of an indexing group or to other interested parties, to be proofread or displayed.

The steps in the indexing procedure are basically these:

1) the creation by each indexer of a master file consisting of lines of assorted subject terms, and the storage of the master file of each indexer in his own disk area,
2) the proofreading and editing of the master files,
3) the processing of the master files:
   a) by using the BASIC computer program that compiles the assorted subject terms and their page references,
   b) by using a high-speed computer sorting program to put the compiled subject terms in alphabetical order and their page references in numerical order,
   c) by using a BASIC computer formatting program to produce a draft of the index, which lists index terms followed by their page references and which also lists modifications of designated terms under those terms, and
4) the proofreading and editing of the draft index.

The procedure requires certain things of those who use it, such as

1) some familiarity with fundamental DECsystem-10 procedures,
2) some knowledge of the DECsystem-10 text editor, called SOS,
3) fair typing ability, and
4) patience with the computer when it breaks down too often.

We will discuss here the first two requirements in some detail. But you should remember that our purpose is not to duplicate the

written instructions and explanations that are already available; it is, instead, to show how to use computer programs that are explained nowhere else but here.

So we will not give details on such steps as logging in, loggging out, and reconnecting to the DECsystem-10, since another, easily obtained manual, the DECsystem-10 Operating System Commands Manual (Revised May 1974; Maynard Mass.: Digital Equipment Corp., 1974), gives that information in its sections on the "LOGIN", "KJOB", and "ATTACH" commands. And we will not explain the entire range of commands used for the SOS text editor or the BASIC programming compiler. We will, however, refer to, and give examples for, some simple SOS commands that a new user can employ effectively until he learns more sophisticated ones that are explained in the SOS "help file", an on-line manual available at most DECsystem-10 installations. (You may order a printout of the manual by this monitor-level command: "PRINT<space>SYS:SOS.HLP<CR>" or "PRINT<space> HLP:SOS.HLP<CR>".) And we will show a bare minimum of commands to the BASIC compiler. A complete discussion of BASIC commands may be found in the DECsystem-10 BASIC Conversational Language Manual (Revised March 1974; Maynard, Mass.: Digital Equipment Corporation, 1974).

Our purpose here is, in fact, not to duplicate any material already available to users of a DECsystem-10 machine. We do hope, though, that by taking advantage of certain resources (such as the DECsystem-10 documentation and the high-speed DECsystem-10 sorting program), and by drawing together in

one place details that are scattered through several sources, we have augmented those resources so as to help you through the more tedious tasks of indexing.

DECsystem-10 MONITOR-LEVEL COMMANDS

The first requirement for the use of the indexing procedure is, as we said, some familiarity with common practices on the DECsystem-10 computer. One of those practices is movement from one level of the computer to another. When you log in on the DECsystem-10 you are at the "monitor" level until you give a monitor-level command that gives you access to another level. While you are at the monitor level, you are given a "dot" (.) prompter; that is, the computer signals you with a dot to show that it is ready for a monitor-level command. Among the commands that you will give in response to the monitor-level prompter to gain access to other levels are "SOS", "R<space>BASIC", and "R<space>SORT".

The first command, "SOS", gives you access to the SOS text editor so that you can type a master file of index entries or edit a rough draft of the formatted index. The command "R<space>BASIC" gives you access to the BASIC programming compiler, which will execute the BASIC programs that process a master file and produce a conventional index from it. The command "R<space>SORT" lets you run the high-speed alphabetizing (sorting) program.

An important point to keep in mind when you give commands at any level of the computer is that you must follow all commands with a "carriage return"; that is, you must strike the

RETURN key. You will not have actually sent the computer a command that you have typed until you have followed the command with RETURN. From here on, we will show with the symbol <CR> all the points at which you are to strike RETURN.

We will also use two other symbols to stand for special terminal keys: ESC, for the ESCAPE (or ALTMODE) key; and CTRL, for the CONTROL key. You will type control characters by holding down the CTRL key (just as you would a SHIFT key) as you strike the control character. For example, to type CTRL C you would hold down CTRL as you strike C. You will use the ESC and CTRL keys when you use the text editor and when you use the BASIC programs.

Three other monitor-level commands that you will find useful in the indexing process are "DIRECTORY", "COPY", and "PRINT". The first one, "DIRECTORY", is what you can use to find out which computer files are stored in your disk area. The computer responds to this command by listing the names and sizes (in computer blocks) of all the files stored. This command helps you to be sure that your master file is in fact stored in your disk area and that the BASIC processing programs that you need are available for use in your disk area.

You will use the second command, "COPY", to bring the BASIC compiling and formatting programs into your disk area so that they can process your master file. The "COPY" command will also be used to collect (merge) the files of several indexers and store them as one file in one disk area. You will need the third command, "PRINT", to order a printed copy of your master

file or of a draft of the index.

A discussion of each of these commands may also be found in the DECsystem-10 Operating System Commands Manual.

USING SOS TO CREATE A MASTER FILE

Now that we have defined the essential monitor-level commands, let us return to the first one, "SOS". This command, as we said, gives you access to the SOS text editor. With SOS you can type new lines into your master file and change lines that are already in it. SOS has two "modes": one (called "Insert") for creating and adding to a file and the other (called "Alter") for changing lines of a file.

Let's start from scratch in creating a computer text file, such as a master file of index entries. At the monitor-level prompter, give the "SOS" command and include the name of the file your want to work on in SOS. Since the BASIC compiling program requires that you call your master file by the name "TERMS.MAS", the command will look like this:

SOS<space>TERMS.MAS<CR>

SOS will respond by recognizing the name of the new file, by putting you in Insert mode, and by giving you the first line number for the file:

```
Creating TERMS.MAS
00100
```

After the line number "00100" type the first line of your master file, ending the line with <CR>.

In Insert mode SOS answers each <CR> with the next line number (greater by 100 than the previous line number) and lets you type in another line of text. You can correct typing errors that you catch on the line you are typing by backspacing (striking the appropriate key on your terminal) and thereby erasing the characters over which you backspace. You can stop inserting new lines by striking the ESC key at the end of the last line you have typed or at the line number give after your last <CR>. When you strike ESC, a "$" will appear on the terminal display at the point where you are in the line. SOS will respond with its own prompter (*), at which you can give any one of a number of commands that are explained in the SOS "help file", some of which we will now describe.

Common SOS commands

Here are some of the more useful SOS commands, with examples of their use.

1) I<line number> - to go into Insert mode after a certain line that you specify.

EXAMPLE: If a group of lines in a file looks like this:

```
00700  Monday 17 October
00800  Tuesday 18 October
00900  Wednesday 19 October
01000  Friday 21 October
```

and a new line should be inserted after line 900, then at the * prompter give the command

I900<CR>

to show that you want to insert a line after line 900. SOS will respond with a new line number, halfway between 900 and 1000, and will let you type the

new line:

00950 <then you type the line><CR>

To show that you want to add one or more lines at the end of the file, give the command

I*<CR>

using the symbol (*) for the last line of the file.

2) P<with or without line number(s)> - to have certain lines of a file displayed at the terminal.

EXAMPLE: If you want to see the next sixteen lines of the file, starting after the line you are on, then at the * prompter give the command

P<CR>

and SOS will respond by showing you those sixteen lines. If you want to see a certain line or a certain range of lines, give the appropriate line number(s), such as:

P110<CR>

or

P110:250<CR>

You can specify a range of lines by typing a colon between the number of the first line of the range and the number of the last line of the range. You can stop the showing of lines by typing CTRL O, by holding down the CTRL key while you type the letter O.

3) D<line number(s)> - to delete a specified line or range of lines.

EXAMPLE: If you want to delete a certain line (say 1150) or a group of lines (say 1360 to 1490), then at the * prompter give the command

D1150<CR>

or

D1360:1490<CR>

and SOS will remove the line(s) you specify and will give you another * prompter.

4) S<string to be replaced>ESC>new string>ESC>line(s)> - to substitute one word or string of characters for another word or string of characters.

EXAMPLE: If you have, say, an abbreviation that you have used throughout your file, and you now want to use only the word that it stands for, then at the * prompter give the "S" command. You could substitute "COMPANY" for "CO." in a certain group of lines (say lines 2150 to 3000) like this:

SCO.<ESC>COMPANY<ESC>2150:3000<CR>

If you want to make this substitution throughout your file, use this form for the range of lines:

↑:*

The "↑" means "the first line of the file", and the "*" stands for the last line. SOS will make the substitution in the range of lines you specify and will show you at the terminal the result of the subsitution.

Some substitutions may have been made that you did not intend. You may, for instance, have had the phrase "FROM AUSTIN TO WACO." in a line of your file. If you had given the last Substitute command that we mentioned (which made the substitution throughout the file), that phrase would now read "FROM AUSTIN TO WACOMPANY", as you would see in the display of results of the substitution. You would then have to correct that phrase by using Alter mode commands in the line that contains "WACOMPANY".

You could have avoided the error in the first place by giving this form of the Substitute command:

S<space>CO.<ESC><space>COMPANY<ESC>↑:*<CR>

With this command you would have singled out the occurrences of "CO." that were preceded by a space. You can stop the display of the results of the substitution with CTRL O, just as you can after the "P" command.

5) F<string><ESC> - to find a line that contains the specified string.

EXAMPLE: If you want to find a line where a certain word or string of characters occurs in your file, then at the * prompter give the "F" command. If you need to add to an index term whose line number you do not know, you can use the "F" command to look through your file for that term. SOS will look for the first occurrence of the term and will show you the line where it occurs.

To add a modification to the index term "COMPUTERS" for certain page references, you could specify that term in your "F" command. SOS would show you the first line that contains the term:

FCOMPUTERS<ESC><CR>
01750 049 COMPUTERS DATA-PROCESSING

If you want to add a modification to "COMPUTERS" for only one or two page references, you could instead search for the page numbers--one at a time--to arrive at the specific occurrences of "COMPUTERS" that you want to modify:

F153<ESC><CR>
02170 153 INFORMATION-RETRIEVAL COMPUTERS
and
F296<ESC><CR>
03240 296 COMPUTERS INFORMATION-ANALYSIS

In general, you will wnat to search from the beginning of the file for one or more occurrences of a string of characters. You can move back to the beginning by giving the "B" command at the * prompter. After SOS shows you the first occurrence of a string, you may want to find later occurrences. To find the later ones, you need only type "F" at the * prompter. SOS will remember the string you told it to find previously and will show you the line where it occurs next.

6) W - to store the file as it stands, while remaining in SOS.

EXAMPLE: After you have typed in or changed twenty lines, day, of your file, leave Insert of Alter mode and at the * prompter give the command

W<CR>

The computer will respond with something like

[DISKBØ: filename]

and will give a new * prompter. You may then resume your work. This command helps you to make sure that the additions and changes you make are not lost even if the system should malfunction.

SOS Alter mode procedures

You can preform what is called "intraline editing" with the SOS Alter mode. Alter mode commands make changes within a line that has already been typed. To correct an error that you have spotted in a certain line, give the "A" command at the * prompter, with the number of the line you want to change:

A1120<CR>

SOS then goes into Alter mode on that line and shows the line number:

01120

You can then move through the line to the point where you want to make a correction. Two ways to move through the line are character-by-character (by striking the space bar once for each character passed) and word-by-word (by striking the "W" key for each word passed). Please note that in Alter mode the marks of punctuation are considered words in themselves. The characters and words will appear at the terminal as you pass over them. You can also use the backspace key to move precisely to the spot where you will make the change.

Once you have moved to that spot, you can either insert new characters or delete some characters that are already there. You can insert characters by typing "I" ("I" will not appear on the terminal display) and then the characters you want to insert. To end this insertion, strike either the ESC key, to leave

yourself in Alter mode on that line, or <CR> , to move yourself out of Alter mode and to another * prompter. You can delete characters by striking "D" (which will not appear on the terminal display) for every character you want to remove from the line. A <CR> ends Alter mode on that line and brings you to another * prompter.

Exiting from SOS

When you have made all the changes in your file that you intend to make, you can send the file to storage in your disk area by exiting from SOS. At the * prompter give the command

EUN<CR>

which moves you out of the SOS level and returns you to the monitor level. To return later to that file to edit it again, give the monitor-level "SOS" command at the dot prompter:

SOS<space>TERMS.MAS<CR>

SOS will respond with

```
Editing TERMS.MAS
*
```

At the * prompter you can give any of the command that we outlined earlier.

When you have completed your editing, you can store the newly edited version of your file by exiting from SOS with the "EUN" command. If you have made serious errors in editing,

however, you will not want to keep the new version but instead will want to store only the original, which would still be stored in your disk area. In this case, you should exit from SOS with the command

EQ<CR>

You can then go back and edit your original with SOS.

You may then want to order a printed copy of your file to have to look over for errors. Often it is easier to read through a file by scanning a printout than by checking it line-by-line at the terminal. To order a printout, give this monitor-level command:

PRINT<space>TERMS.MAS<CR>

As we said earlier, the SOS commands and procedures outlined here are the simplest ones. A new SOS user can manage well enough with them. But other, more powerful commands are available; and those who will be using SOS more than three or four times will do well to learn all the shortcuts they can from the SOS "help file".

REQUIREMENTS FOR THE MASTER FILE

Now that we have discussed how to make the master file, we should say something about what to include in it. The compiling and formatting programs will expect to find certain items in the lines of your file. If you include those items and use some substitutes for certain marks of punctuation, the programs will run correctly and your editing of the result will be simple.

These are the limitations on what you may have in the lines of your file:

1) Each line can contain no more than 140 characters, counting spaces and punctuation.

2) Each line has this format:
<page no.><space><index terms separated by spaces>

3) All page numbers should consist of three digits if any one of them is greater than 99.

4) An index term consisting of more than one word must be hyphenated, since spaces are used to separate index terms from each other.

5) An index term (a subject term or its modification) should consist of no more than about 70 characters, counting the hyphens.

If you have an index term that requires a modification, you may type that index term, then a number sign, and then the modification. Here again, if either the index term or its modification has two or more words, you must hyphenate those words.

An index term with its modification, will look like this in a line of your master file:

COMPUTER-TECHNOLOGY#RECENT-RESEARCH

The formatting program will list the modification "RECENT-RESEARCH" under the index term "COMPUTER-TECHNOLOGY" and will attach to the modification the appropriate page reference, like so:

COMPUTER-TECHNOLOGY
RECENT-RESEARCH, <page number>

In general, you will have lines that begin with page references, which are followed by index terms with or without modifications. The index terms will be separated by spaces from the other index terms on the same line. Lines of your master file could look like these:

079<space>COMPUTER-TECHNOLOGY#RECENT-RESEARCH
121<space>INFORMATION-RETRIEVAL<space>COMPUTERS
156<space>COMPUTERS<space>DATA-BANKS<space>DATA-STORAGE<space>STORAGE-MEDIA<space>INVERTED-FILES
174<space>COMPUTER-TECHNOLOGY<space>DATA-STORAGE

The formatting program would produce these entries in the index:

COMPUTER-TECHNOLOGY, 174
RECENT-RESEARCH, 079
COMPUTERS, 121, 156
DATA-BANKS, 156
DATA-STORAGE, 156 174
INFORMATION-RETRIEVAL, 121
INVERTED-FILES, 156
STORAGE-MEDIA, 156

You will do well to choose the shortest possible index terms and to have the lowest possible posting densities (by choosing the most specific index terms possible). We will explain the technical reason for these words of counsel when we discuss the formatting program. Until then, you can probably already agree that it is easier to correct indexing that is too specific than it is to supplement indexing that is too general.

Once the formatted index has been produced, you can use SOS to substitute spaces for hyphens that you had to type between the words of index terms. But if you intend to have some hyphenated words as index terms, or if you want commas within index terms, you must use other marks of punctuation instead.

Suggested punctuation conventions

Here are some ideas on which marks of punctuation to use when you want hyphens and commas within index terms. We suggest these conventions only because they seem easier to type than the other possibilities. You may decide to use punctuation other than the marks shown here; and since the compiling program acts only on spaces in the lines of your master file, it will not distinguish among the marks of punctuation you use, except for number signs.

You will find it to your advantage, however, to use conventions such as these so that your final editing of the formatted index is as rapid as possible. The punctuation conventions that we suggest are:

1) FOR <space> USE <hyphen>
2) FOR <hyphen> USE <semicolon>
3) FOR <comma><space> USE <semicolon><semicolon>

At the end of the indexing procedure we will discuss the SOS Substitute commands that correspond to these conventions.

Eliminating errors of spacing in master file lines

Once the master file is complete, you should order a printout and look it over once or twice before you process the file, using SOS to correct errors that you find. A common error is to have two or more spaces between elements in a line of the master file. This error is hard to detect, especially if you must look for it through a file of hundreds of lines. And yet the compiling program will not properly process the master file if it finds even one line with two spaces in a row. All things considered, it appears more efficient--it will save processing time in the long run--to have your master file conform to the program's present requirements than to have the program continually look especially for occurrences of overspacing and then adjust itself accordingly.

The present procedure seems more efficient because it is easy, using SOS, to eliminate all occurrences of overspacing. All you need to do is to bring the master file into SOS by the monitor-level command

SOS<space>TERMS.MAS<CR>

and then give a Substitute command that will correct the errors

without your having to look for them, such as this one:

S<space><space><ESC><space><ESC>↑:*<CR>

You thus substitute a single space for every occurrence of a pair of spaces in your file.

Another easily made error is to have a space at the end of a master file line. This error will also prevent the compiling program from processing the master file. But unlike the first error, this one is impossible to detect by scanning lines of a printout and is nearly impossible to find by surveying the lines of a file at the terminal. As a time-saving precaution, then, you should, before you run the compiling program, give the SOS Substitute command that will correct the error without your having to look for it:

S<space><CTRL E><CTRL X><ESC><ESC>↑:*<CR>

You thus specify that you are substituting nothing for every space found at the end of a line.

PROCESSING THE MASTER FILE

Using the BASIC compiling program

If you are reasonably sure (you can never be completely sure, so don't worry) that your master file is free from error, then you can go to the next step in the procedure: the compiling of the master file. You will use a BASIC program, ARCHIV.BAS, to pair each index term with its page reference.

If the program is not already in your disk area, you can call it in with this command at monitor level:

COPY<space>=ARCHIV.BAS[P,PN]<CR>

(In this command "[P,PN]" stands for the project-programmer number, such as "[2313,75]", for the disk area in which the program is stored.) After the computer gives you another dot prompter, you can proceed to run the program. You enter the BASIC level of the computer with this monitor-level command:

R<space>BASIC<CR>

The computer responds in a new way:

```
READY, FOR HELP TYPE HELP.
>
```

The ">" is the prompter that you get at the BASIC level. At the prompter type

OLD<space>ARCHIV<CR>

and when BASIC gives you another prompter, type

RUN<CR>

BASIC will then show a heading, telling the program's name, the time, and the date. And the operation will seem to cease for a while. Actually the program ARCHIV.BAS will automatically begin to process your TERMS.MAS file and will show you at the terminal the result of the compilation. We suggest that you simply have patience until the display of results begins. Once the display starts, you need not read all the way through it; you can save yourself time by typing CTRL O, just as we discussed earlier for the SOS "P" and "S" commands, and then waiting for the "TIME" line that the program will give you when it is finished.

Once the program run is complete, exit from the BASIC level by typing CTRL C.

You now have a file that the program has created in your disk area and has called "RESULT.BAS". This file is the raw form of the formatted index and is hard to read. You will process it further in the next steps to arrive at a readable index.

Now if the program did not show the result of its compilation but instead gave an "error message", you will need to exit from BASIC (with CTRL C) and make corrections in your master file. As a rule, you should assume that the program failed not through its own fault but instead because of a flaw in your master file. If the error message shown is

? MID$ ARGUMENT OUT OF BOUNDS IN LINE . . .

then the master file probably contains at least one line with a spacing error. You should go back into SOS with your file and give one or both of the Substitute commands that we discussed on pages 20-21 to get rid of the errors the program has found.

Up to this point we have discussed what each indexer in a project group can do on his own. From here one, the steps described will be followed by one person in the group, who will carry out the sorting and formatting.

Using the DECsystem-10 SORT program

The next step in the indexing procedure is the merging and alphabetizing of the RESULT.BAS files on the project group members. To merge the RESULT.BAS files, give this monitor-level command:

```
COPY<space>=RESULT.BAS, RESULT.BAS[P,PN],
   RESULT.BAS[P,PN], . . . , RESULT.BAS[P,PN]<CR>
```

(In this command "[P,PN]" stands for the project-programmer numbers that correspond to the disk areas of the members of the group.) When the computer gives another dot prompter, copies of the files in the separate disk areas will have been merged in your disk area.

You now alphabetize the newly merged RESULT.BAS file by using the DECsystem-10 SORT program. To run SORT, give this monitor-level command:

```
R<space>SORT<CR>
```

SORT then gives you its own prompter (*). At the prompter type this line:

RESULT.BAS=RESULT.BAS/K:1:80/R:80<CR>

Be sure that you have typed the colons in the correct places.

What you are saying in this command to SORT is that you want the sorted file to have the name "RESULT.BAS" and that the file you want sorted (to the right of the equal sign) is also called "RESULT.BAS". You then are specifying (by "/K:1:80") that you want the lines of the file (each of which contains an index term paired with its page reference) alphabetized, starting at the first character of each line and ending at the eightieth, if the line happens to have that many characters. Then you are telling SORT (by "/R:80") that your longest line has no more than 80 characters.

After you give that command, SORT will respond with several lines of information and then will give you another * prompter. At that point, exit from SORT and return to monitor level with CTRL C. The file RESULT.BAS is now ready to be formatted to look like a conventional index.

Using the BASIC formatting program

To carry out the formatting you must first bring the formatting program into your disk area, if it is not there already, with this monitor-level command:

COPY<space>=FORMAT.BAS[P,PN]<CR>

When the computer gives you another dot prompter, move to the BASIC level with this monitor-level command:

R<space>BASIC<CR>

At the BASIC prompter (>), type this command:

OLD<space>FORMAT<CR>

At the next BASIC prompter, type the command

RUN<CR>

The program begins with a heading similar to that of ARCHIV.BAS. It will not show the result but instead will finish with a "TIME" line after a short while. Here again, assume that the program is running correctly even though its operation appears to have stopped after the heading. When the program gives you the "TIME" line, exit from BASIC with CTRL C.

The program will have created a new file in your disk area and will have called it "INDEX.BAS". This file is your formatted index. The lines of this index will contain no more than 80 characters, and this limitation imposed by programming in BASIC brings us to an important point about your indexing policy.

Since the lines of your formatted index can be no more than 80 characters long, you should keep your index terms and their modifications as short as you can and keep your posting densities moderate. By doing so, you will insure that your index lines will not be too long for the formatting program to

handle as it gathers together all the page references for an index term or modification and adds them to the line that begins with that term or modification.

If a line containing an index term and its page references is longer than 80 characters, you will see an error message during the run of FORMAT.BAS. That message will be a signal to you that your posting density is too high for some unspecified term.

You can find out which term it is by calling your RESULT.BAS file into the SOS text editor and giving an "F" command to tell SOS to show you all occurrences of a certain index term. For example, if you suspect that the term "computers" may have too many page references attached to it, you can start by giving the monitor-level command

SOS<space>RESULT.BAS<CR>

and at the * prompter giving this command:

FCOMPUTERS<ESC>,50<CR>

In response to this command SOS will show all the occurrences of the string of characters "COMPUTERS"--up to fifty occurrences, if your file has that many. If this index terms turns out to have an inordinate number of page references, you could add modifications to it at several points in your file. By doing so you would make your file conform to the limitations of the BASIC program, and you would also, of course, make your indexing more specific.

Using SOS to replace the punctuation conventions

The index is not quite in final form at this point. The step that remains is for you to edit the punctuation in the index entries. You will use SOS to substitute the intended punctuation for the punctuation conventions that we suggested earlier on pages 19-20.

You can call the file INDEX.BAS into SOS with this monitor-level command:

SOS<space>INDEX.BAS<CR>

Then at the * prompter give this Substitute command:

S;;<ESC>,<space><ESC>↑:*<CR>

After that substitution (of comma-and-space for a pair of semicolons) is complete, give another Substitute command:

S-<ESC><space><ESC>↑:*<CR>

After the second substitution (of a space for every hyphen) is complete, give the last Substitute command (changing all remaining semicolons to hyphens):

S;<ESC>-<ESC>↑:*<CR>

ORDERING A PRINTED COPY OF THE INDEX

Once the three substitutions have been made, the file should look quite presentable. It will certainly be ready to be printed on a high-speed line printer, for proofreading and possible reworking (using SOS Alter mode and other editing commands). You can order three copies, say, of a printout of the formatted and edited index by giving this monitor-level command:

PRINT<space>INDEX.BAS/COPIES:3<CR>

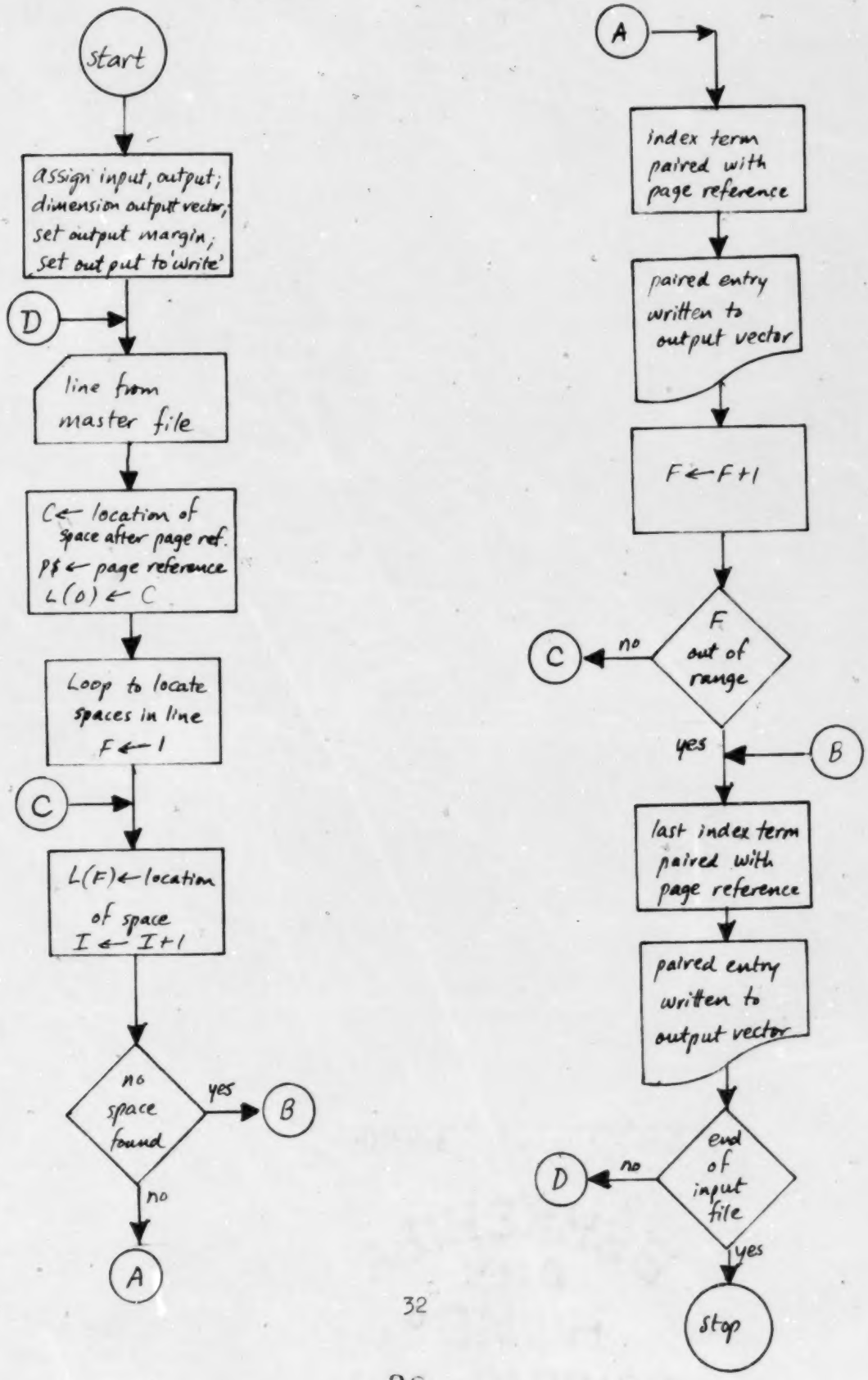
BASIC program ARCHIV.BAS, with annotations.

```
00100   FILES TERMS.MAS, RESULT'          Designate input and
00150   '                                       output files.
00200   DIM D$(5000)'                     Dimension vector of
00250   '                                       output lines.
00300   MARGIN 80, #2:80'                 Set margins of display
00350   '                                       and output.
00400   SCRATCH #2'                       Set output file to write
00450   '                                       mode.
00500   LINPUT #1, T$'                    Read a line from TERMS.MAS
00550   '                                       (input).
00600   C = INSTR (T$," ")'               Assign variable "C" the
00700   '                                       value of
00800   '                                       position after
00850   '                                       page reference.
00900   P$ = "*" + LEFT$(T$,C-1)'         A substring of input
00950   '                                       line consisting
01000   '                                       of separator (*)
01050   '                                       and page
01075   '                                       reference.
01200   L(0) = C'                         Assign the 0 slot of the
01250   '                                       locations
01275   '                                       vector (L)
01300   '                                       the value for
01350   '                                       the position of
01400   '                                       the space after
01500   '                                       the page reference.
01600   FOR F = 1 TO LEN (T$)'            Begin loop, to find all
01650   '                                       spaces in input
01700   '                                       line after page
01750   '                                       reference.
01800   L(F) = INSTR(L(F-1) + 1, T$, " ")'  Locate spaces and
01850   '                                       store values for
01900   '                                       locations in
01950   '                                       vector L.
02100   I = I + 1'                        Set counter for location
02150   '                                       in vector D$.
02200   IF L(F) = 0 THEN 2900'            If no space found, skip
02250   '                                       out of loop.
02300   D$(I) = MID$(T$, L(F-1) + 1, (L(F) - 1) - (L(F-1))) + P$
02350   '                                 In line 2300, read
02375   '                                       identified index
02400   '                                       terms and their
02450   '                                       page references
02500   '                                       into D$.
02600   PRINT D$(I)'                      Show line of D$ at
02650   '                                       terminal.
02700   PRINT #2, D$(I)'                  Write line of D$ to
02750   '                                       output file.
02800   NEXT F'                           Repeat loop.
```

ARCHIV.BAS continued

```
02900   DS(I) = MIDS(IS, L(F-1) + 1) + PS'  Read last index
02950   '                                       term in input
03000   '                                       line into DS.
03050   PRINT DS(I)'                        Show line of DS at
03100   '                                       terminal.
03150   PRINT #2, DS(I)'                    Write line of DS to
03200   '                                       output file.
03250   IF END #1 THEN 3550'                Check for end of input
03300   '                                       file.
03350   GO TO 500'                          If not end of input
03400   '                                       file, then read
03450   '                                       in another input
03500   '                                       line.
03550   END
```

Flowchart of ARCHIV.BAS



BASIC program FORMAT.BAS, with annotations.

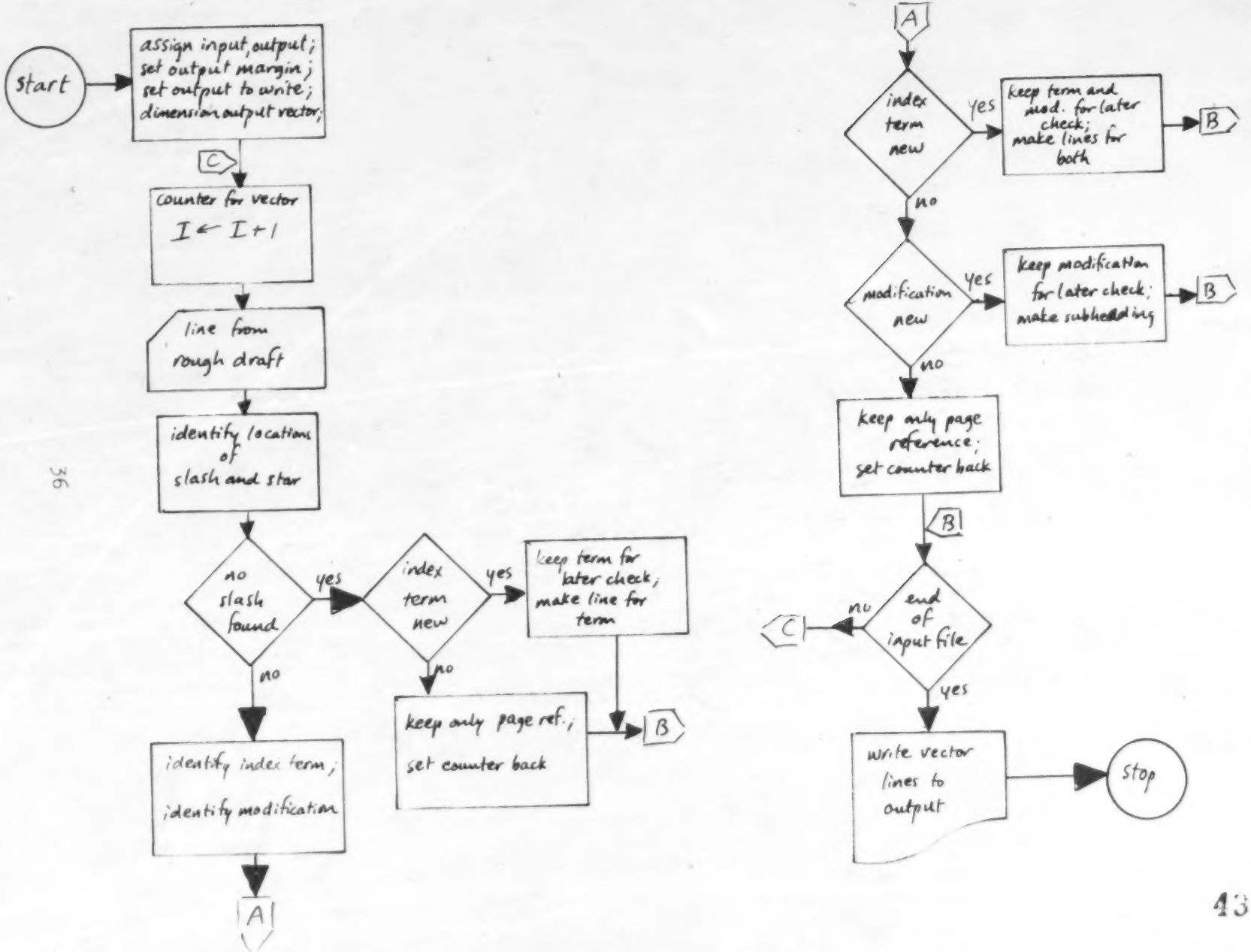
```
00100   FILES RESULT, INDEX
00200   MARGIN #2:80
00300   SCRATCH #2
00400   DIM RS(5000)'                       Dimension vector to
00500   '                                           hold lines of
00550   '                                           formatted index.
00600   I = I + 1
00700   LINPUT #1, RS(I)
00800   S = INSTR(RS(I), "#")'              Check input line for "#",
00900   '                                           showing a
01000   '                                           modification in
01050   '                                           line.
01100   Z = INSTR(RS(I), "*")'              Note location of "*"
01200   '                                           separator in
01250   '                                           input line.
01300   IF S = 0 THEN 5700'                 If no slash, then avoid
01400   '                                           manipulation of
01500   '                                           modification.
01600   T1$ = LEFT$(RS(I), S-1)'            T1$ is index term.
01700   T2$ = MID$(RS(I), S+1, Z-(S+1))'    T2$ is index term's
01800   '                                           modification.
01900   IF T1$ <> K1$ THEN 4500'            If index term has not
02000   '                                           been encountered
02100   '                                           before, store
02200   '                                           it for later
02250   '                                           reference.
02300   IF T2$ <> K2$ THEN 3800'            If index term has been
02400   '                                           encountered
02500   '                                           before, but its
02600   '                                           modification has
02700   '                                           not, store
02725   '                                           modification
02750   '                                           for later
02775   '                                           reference.
02800   RS(I-1) = RS(I-1) + ", " + MID$(RS(I), Z+1)
02900   '                                   In line 2800, if both index
03000   '                                           term and
03050   '                                           modification
03100   '                                           have been
03150   '                                           encountered
03200   '                                           before, keep
03250   '                                           only the page
03300   '                                           reference.
03400   I = I - 1'                          Reset the counter for the
03500   '                                           location in the
03600   '                                           vector of index
03650   '                                           lines.
03700   GO TO 7700'                         Go to end-of-file check.
03800   K2$ = T2$'                          Store modification for
03900   '                                           later reference.
```

FORMAT.BAS continued

```
04000   RS(I) = SPACE$(8) + I2$ + ", " + MID$(RS(I), Z+1)
04100   '                                  In line 4000, make
04200   '                                        indented
04250   '                                        subheading out
04300   '                                        of modification.
04400   GO TO 7700'                        Go to end-of-file check.
04500   K1$ = I1$'                         Store index term for
04600   '                                        later reference.
04700   K2$ = I2$'                         Store modification for
04800   '                                        later reference.
04900   RS(I+1) = SPACE$(8) + I2$  + "," + MID$(RS(I), Z+1)
05000   '                                  In line 4900, make
05100   '                                        indented
05150   '                                        subheading out of
05200   '                                        modification.
05300   RS(I) = I1$'                       Make line for index
05350   '                                        term.
05400   I = I + 1'                         Increase counter for
05450   '                                        location in
05500   '                                        vector of index
05550   '                                        lines.
05600   GO TO 7700'                        Go to end-of-file check.
05700   IF MID$(RS(I), 1, Z-1) <> K1$ THEN 7000
05800   '                                  In line 5700, if index
05850   '                                        term is
05900   '                                        unmodified,
06000   '                                        check to
06050   '                                        see if it was
06100   '                                        encountered
06150   '                                        before.
06200   RS(I-1) = RS(I-1) + ", " + MID$(RS(I), Z+1)
06300   '                                  In line 6200, if index
06400   '                                        term was
06450   '                                        encountered
06500   '                                        before, keep
06550   '                                        only the page
06600   '                                        reference.
06700   I = I - 1'                         Reset the counter for the
06800   '                                        vector of index
06850   '                                        lines.
06900   GO TO 7700'                        Go to end-of-file check.
07000   K1$ = MID$(RS(I), 1, Z-1)'         If index term was not
07100   '                                        encountered
07200   '                                        before, store
07250   '                                        it for later
07300   '                                        reference.
07400   RS(I) = MID$(RS(I), 1, Z-1) + ", " + MID$(RS(I), Z+1)
07500   '                                  In line 7400, make an
07600   '                                        index line for
07650   '                                        the index term.
07700   IF END #1 THEN 8000'               Check for end of file.
07800   GO TO 600'                         If not end of file,
07850   '                                        repeat
07900   '                                        algorithm.
```

FORMAT.BAS continued

```
08000    FOR X = 1 TO I'                    Begin loop to write
08100    '                                          index lines from
08150    '                                          vector RS
08200    '                                          to output file.
08300    PRINT #2, RS(X)
08400    NEXT X
08500    END
```



Flowchart of FORMAT.BAS